AD_________________

Award Number: W81XWH-06-1-0096

TITLE: Development of an Amendment to X3D to Create a Standard Specification of Medical Image Volume Rendering, Segmentation, and Registration

PRINCIPAL INVESTIGATOR: Sandy Ressler
Mike Aratow, M.D., FACEP

CONTRACTING ORGANIZATION: Web3D Consortium
San Francisco, CA 94104

REPORT DATE: November 2006

TYPE OF REPORT: Final

PREPARED FOR: U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

DISTRIBUTION STATEMENT: Approved for Public Release;
Distribution Unlimited

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE
01-11-2006

2. REPORT TYPE
Final

3. DATES COVERED
20 Oct 2005 – 20 Oct 2006

4. TITLE AND SUBTITLE
Development of an Amendment to X3D to Create a Standard Specification of Medical Image Volume Rendering, Segmentation, and Registration

5a. CONTRACT NUMBER

5b. GRANT NUMBER
W81XWH-06-1-0096

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Sandy Ressler
Mike Aratow, M.D., FACEP

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Web3D Consortium
San Francisco, CA 94104

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
There is currently no standard file format for representation of three dimensional (3D) medical imaging data. Extensible 3D(X3D) is an International Standards Organization(ISO)-ratified, freely available standard that defines a runtime system and delivery mechanism for 3D graphics on the World Wide Web. The Web3D Consortium, which administers X3D, has developed a draft extension to X3D for a volume rendering, registration and segmentation component to define a file format and display of 3D medical imaging data. A formal ISO working project has been initiated to begin the process of ISO ratification of this extension.

15. SUBJECT TERMS
X3D, Web3D, Consortium, ISO, volume rendering, standard, segmentation, registration

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			USAMRMC
U	U	U	UU	41	19b. TELEPHONE NUMBER *(include area code)*

Table of Contents

Introduction

The goal of this project was to initiate development of an extension to the Extensible 3D (X3D) Specification: the recently approved new ISO standard for representing 3D graphics on the World Wide Web, for displaying and manipulation of volumetric data.. Before this project, the X3D specification did not contain extensions for volume rendering, segmentation or registration. This additional functionality is key to adoption of this standard in the medical community because it adds significant benefits and an impetus to innovation in such areas of medicine as professional education (e.g. as an adjunct to teaching modalities in anatomy), patient education (e.g. informed consent of complex surgical procedures) and surgical planning (e.g. visualization for complex surgical procedures).

The project began with an in-depth analysis of the both the technical and user needs of this extension. From this analysis, new X3D nodes were identified along with their associated variables. These new nodes were then written in the document format structure specified by the ISO specification requirements. This document—the Volume Rendering Component—is being vetted through the Web3D Consortium's approval process. The Consortium is also seeking comments from selected members of the computer graphics community outside of the membership. During this process, specification work (standards editing) has been conducted to produce a specification that can meet ISO's rigorous requirements. The document resulting from this process is now being prepared for submittal as an official ISO project. The Web3D Consortium has also begun communications with the DICOM standards group to ensure a smooth interface with this current international 2D standard for medical imaging.

Body

The proposers identified the following tasks as the key components of this project:

1) Identification of new X3D nodes.
2) Initiation of a formal ISO working project.
3) Editing of the new X3D nodes into a formal standard.

Task 1. Identification of New X3D Nodes

In order to maximize the utility of X3D for the medical community, the specification needed to be extended to accommodate "voxel" type representations. This task required specific details of a voxel node and any supporting nodes, and would need support for large data culling as well as support for segmentation and registration. The Volume Rendering Component is designed for implementation on consumer hardware using shaders. Hardware specifically designed for volume rendering is not required.

The Web3D Medical Working Group has determined that this new component will need two new node types and twelve new nodes to accommodate voxel rendering,

large data culling, segmentation and registration. The Working Group identified two new node types that would form the basis for the Volume Rendering Component:

1) *X3DVolumeNode*.
2) *X3DVolumeRenderStyleNode*.

X3DVolumeNode is the base type for all X3D nodes that describe the volumetric data to be rendered. It sits at the same level as the polygonal *X3DShapeNode* within the scene graph structure, but defines volumetric data rather polygons. The new X3D nodes governed by this node type are:

1) *SegmentedVolumeData*
2) *VolumeData*

X3DVolumeRenderRenderNode is the base type for all node types which specify a specific visual rendering style to be used. The end user is able to turn on and off volume rendering of specific parts of the volume without needing to add or remove style definitions from the volume data node. The new X3D nodes governed by this node type are:

1) *BoundaryEnhancementVolumeStyle*
2) *CartoonVolumeStyle*
3) *CompositeVolumeStyle*
4) *EdgeEnhancementVolumeStyle*
5) *ISOSurfaceVolumeStyle*
6) *OpacityMapVolumeStyle*
7) *SilhouetteEnhancementVolumeStyle*
8) *StippleVolumeStyle*
9) *ToneMappedVolumeStyle*

These nodes include most common rendering styles used in medical imaging. The specification also allows the user to employ a composite style enabling multple styles to be used together to highlight different segments. A common example is to render bone and blood vessels with one shader and the surrounding skin with another.

The *OctTree* node was created using existing X3D node types. It allows for the definition of multiresolution data sets that resolve using octants of volume.

The specification document for the Volume Rendering Component is provided in Appendix A.

<u>Task 2. Initiation of a Formal ISO Working Project</u>

As new nodes were defined, they were realized within the defined forms of X3D profiles and components. In addition to those definitions, a new work item proposal to ISO was prepared and added to the X3D specification in a new component called

Volume Rendering. This component will be part of the full profile in X3D Amendment 2.

Future work may define a specific profile for the medical market. This new profile will require further research to identify specialized features required from X3D to meet the medical market's needs.

The ISO work item proposal is included as Appendix B.

Task 3. Editing of the New X3D Nodes into a Formal Standard

The Volume Rendering Component was approved by the Web3D Medical Working Group in December 2005. It was edited into a coherent ISO standards document by Richard Puk (a qualified ISO editor) and prepared for inclusion in the X3D Amendment Two process. Amendment Two is being submitted to ISO for ratification on March 1, 2006. It will enter the ISO process as a committee draft and will take approximately one year before it becomes an official ISO standard.

To facilitate acceptance of X3D as a relevant pathway for medical imaging, the Web3D Consortium began interaction with the DICOM standards body. A presentation was made to Working Group 17 of DICOM on February 13, 2006 in San Diego, CA. This presentation is being followed up with further communications with other relevant working groups within DICOM.

Key Research Accomplishments

The proposers accomplished the following during this project:

1) Identified user requirements for a volume rendering component.
2) Identified main technical issues for incorporating volume rendering in the X3D specification.
3) Identified key nodes and fields for a volume rendering component for the X3D specification.
4) Initiated of a formal ISO working project for the X3D volume rendering component.
5) Begun interaction with DICOM standards group.

Reportable Outcomes

The proposers can report the following:

1) The Volume Rendering Component specification was completed.
2) The Volume Rendering Component specification was approved by the Web3D Medical Working Group and was submitted to the Web3D Consortium Board of Directors for final approval.

3) The Volume Rendering Component specification was approved by the Web3D Board of Directors and was submitted to ISO for approval with X3D Amendment Two.

4) Proposers completed a new work item proposal to ISO that was added to the X3D specification in a new component called Volume Rendering. This component will be part of the full profile in X3D Amendment Two.

Conclusion

The Volume Rendering Component for the X3D specification provides an open standard for describing volume rendering-based data generated from medical imaging devices such as MRIs, CAT and PET scanners. An important feature to this component is the XML encoding component of the X3D standard that facilitates the distribution of this data over networks, and allows one to incorporate metadata with the volume data. This feature will have tremendous impact on storage and distribution of patient records, medical education and research. The Volume Rendering Component was completed and submitted to ISO for approval as a part of the X3D standard.

A standard file format for three-dimensional medical images has several important implications. First and foremost, it provides a common language for organizations in the government, private and commercial sectors with which to communicate this type of data. This allows more rapid turnaround times in interactions between organizations and fewer errors since no data conversion is needed, and economic savings for the end user since all vendors must interoperate with the standard, fostering competition and lower cost of ownership. A standard file format also stimulates innovation since independent developers can focus their energy on applications instead of 3D data expression. Moreover, they are assured of a guaranteed installed base of end users who recognize and use the standard. Finally, a standard will allow for a significant increase in accessibility of 3D medical images to professionals, students and lay persons where currently such groups must view such images on specialized radiological workstations that have limited access. With a standard file format, these images will be able to be displayed on any home computer.

However, the successful completion of the specification writing process is only the first step in creating a specification that will meet the specialized needs of the medical community. Indeed, the Web3D Consortium is seeking to refine and extend X3D and its Volume Rendering Component to better serve this sector. To this end, the Consortium is seeking funding to continue its work in this area.

The Web3D Consortium's standardization process requires two implementations of its specifications before final ISO ratification. Thus the Consortium will need to encourage two developers to provide software implementations of the Volume Rendering Component by March 2007. Yumetech, Inc. and Media Machines have

performed feasibility testing for both OpenGL and DirectX. The two browser implementers provided feedback on whether they could incorporate the specification as written into their software. They provided assurance that the new component can be implemented using current consumer hardware, thereby ensuring that the standard is widely adopted.

Another requirement for the new component will be the development of a robust conformance suite to insure that all software implementations of the specification are interoperable. This suite will provide tests for all the nodes and fields of the Volume Rendering Component. Software vendors will have to show that their software passes all these tests before they can claim that they are conformant. The Web3D Consortium will schedule conformance events to allow vendors to test their software and receive certification.

As users begin to adopt the Volume Rendering Component, the Medical Working Group will need to monitor how it is being used and determine whether the medical sector requires the creation of a new X3D profile. In the X3D Specification, a profile can be targeted at a specific market application. These market-specific profiles define what X3D capabilities are required to meet the market requirements. Some markets require the majority of the X3D nodes while others require only a small subset. One example is the CAD Distillation Format (CDF) developed by the CAD Working Group. The CDF uses a small subset of the specification to foster interchange from CAD systems to visualization systems. The Medical Working Group will need to perform a market analysis in order to customize the X3D specification accordingly.

The Medical Working Group also recognizes that it will need to actively promote X3D and the Volume Rendering Component to facilitate their adoption. The group plans a number of activities to achieve this goal. These activities include public presentations at relevant conferences and the publication of academic papers describing medical applications based on X3D. Although the Web3D Consortium performs general marketing for all Web3D standards, each working group is responsible for promoting their work in their specific vertical market. Because of the size and scope of the medical field, the Medical Working Group will require more funding for the academic promotion of its work.

It must be noted that all the work of both the Web3D Consortium and the Medical Working Group—including the X3D Specification and the Volume Rendering Component—is royalty-free. The Consortium requires intellectual property rights (IPR) statements from its contributors that specify no intellectual property encumbered technology may be included in their contributions to the specification. In this way, all its standards can be widely adopted.

Appendices

Appendix A

3D ANYWHERE

Extensible 3D (X3D)
Part 1: Architecture and base components

Volume Rendering Component (Extension Proposal)

1. Introduction

1.1 Name

The name of this component is "VolumeRendering". This name shall be used when referring to this component in the COMPONENT statement (see ISO FDIS 19775-1:200x 7.2.5.4 Component statement).

1.2 Overview

This clause describes the Volume Rendering component of this part of ISO/IEC 19775. This includes how volumetric data sets are are specified and how they are rendered . Table 1 provides links to the major topics in this clause.

Table 1 — Topics

Table 2 — Volume Rendering component support levels

2 Concepts

2.1 Overview

Volume Rendering is an alternate form of visual data representation compared to the traditional polygonal form used in the rest of this specification. Where polygons represent an infinitely thin plane, volume data represent a three dimensional block of data. When polygonal data representing a volume in space is sliced, such as with a clipping plane, there is empty space. In the same situation volumetric data will show the internals of that volume.

There are many different techniques for rendering volumetric data - plane slicing, use of real time shaders, ray tracing etc. This component does not define the technique used to render the data, only the type of visual output needed. For example, a simple set of opacity data has one representation, while segmented data sets have another. In order to implement some of the higher complexity representations, the implementor may need to use a more complex technique than the simpler representations, though it is not required. Each of the rendering nodes will represent the visual output required, not the technique used to implement it.

2.2 Representing Volumetric Data

2.2.1 Registration and Scaling

Volumetric data represents volume information that typically comes from the real world: for example human body scans or finite element analysis of an engine part. The volumetric data is typically part of a larger environment space and thus needs to be located within that space, so that volumes for different parts (eg arm and leg of a single human) may be presented in a spatially correct manner. Typically volumes are not a unit cube in size, so extra dimensional information must be provided with the volume to indicate its true size in the local coordinate system. The volume data shall be effected by the normal transformation heirarchy, including scaling and shearing.

2.2.2 Data Representation

Volume rendering requires providing data in a volumetric form. This component uses the 3D texturing component (See ISO/IEC 19775-1 PDAM-1 3D-Texturing) to represent the raw volume data, but without rendering that data directly onto polygonal surfaces. Volumetric rendering may make use of multiple 3D textures to generate a final visual form.

Data may be represented using between 1 and 4 colour components. How each colour component is to be interpreted as part of the rendering shall be defined for each node. Some nodes may require a specific minimum number of components, or define that anything more than a specific number will be ignored. Providing extra data may not be helpful to the implementation.

An implementation is free to provide whatever data reduction techniques are desired under the covers. Explicit volume data representations are provided in the *OctTree* node that allows the user to describe progressively more detailed volumetric data. When the user presents data in this form, it shall be followed as the required rendering technique. However, within a specific volume data representation, the implementation may also perform its own optimisation techniques, for example automatic mipmapping.

Volume visualisation data sets are not required to be represented in sizes that are powers of two. Implementations may need to internally pad the texture sizes for passing to the underlying rendering engine, but user-provided content is not required to do this.

2.2.3 Segmentation Information

The volme data may optionally represent segmented data sets. Doing so requires representing the data in a slightly different manner than a standard volume data set, so a separate node is used. Segmentation data takes the form of an additional volume of data where each voxel represents a segment ID value in addition to other values represented in each voxel. The segmentation information is used by the rendering process to control how each voxel is to be rendered. It is not unusual to use segmentation information to render each segment identifier with a different style - for example bone using ISO surfaces, skin using tone shading etc.

2.2.4 Tensor Representation

This specification does not explicitly handle or represent tensor data. Tensor information may be rendered using the techniques in this profile,

even though no direct data is being transmitted. It is recommended that if an application needs to know about the existance of tensor data, that the metadata capabilities of the specification be used.

2.2.5 Visual Representation

Volumetric data is typically given as a rectangular block of information. Turning that into something meaningful where structures may be discernable is the job of the rendering process. However, there is not a one-size-fits-all approach to volume rendering. A technique that is good for exposing structures for medical visualisation may be poor for fluid simulation visualisation.

To allow for these different visual outputs, this component separates the scene graph into two sets of responsibilities - nodes for representing the volume data, and nodes for rendering that volume data in different ways. In this way, the same rendering process may be used for different sets of volume data, or varying rendering styles may be used to highlight different structures within the one volume.

Many rendering techniques map the volume data to a visual representation through the use of another texture known as a Transfer function. This secondary texture defines the colours to use, acting as a form of lookup table. Transfer functions can be defined in 1, 2 or 3 dimensions. As X3D does not define a 1-dimensional texture capability, this can be simulated through the use of a 2D texture that is only 1 pixel wide.

2.3 Interaction with Other Nodes and Components

2.3.1 Overview

Volumetric rendering requires a completely different implementation path to traditional polygonal rendering. The data represents not only surface information, but also colour and potentially lighting. As such, volume rendering occupies the space in the renderable scene graph that is a *X3DShapeNode* rather than as individual geometry or appearance information.

2.3.2 Lighting

Volumetric rendering is not required to follow the standard lighting equations for this specification. Many techniques will include the ability to self-light and self-shadow using information from the parent scene graph (light scoping etc).

The volume data is rendered using one or more rendering styles. Each style defines its own lighting equation. Many of these involve non-photorealistic effects. Each style will present it's own lighting equation on how to get from the underlying voxel representation to the contributed output colour. The following are some common terms that will be found in the lighting equations:

- O_v: The initial opacity of the object prior to the use of this style. This opacity may include a previously-calculated transfer function evaluation.
- O_g The output opacity of the object resulting from evaluating this style.
- C_v: The initial colour of the object prior to the use of this style. This opacity may include a previously-calculated transfer function evaluation.
- C_g The output colour of the object resulting from evaluating this style.
- Δf: The normalised value gradient of the voxel. This is the rate of change of the value relative to the values in neighbouring voxels.
- **V**: The vector from the viewer's position to the voxel being evaluated, in the local coordinate space of the volume data
- **n**: The local surface normal. This may be provided by the user through another 3D texture that contains a surface normal for each voxel or internally calculated through algorithmic means.
- $\mathbf{L}_i$ Light direction vector from light source *i*. Typically involves a summation over all light sources affecting the volume.

2.3.3 Geometry

The volumetric rendering nodes are a leaf node in the renderable tree. Volumetric nodes may exist as part of a shared scene graph with DEF/USE, though it is expected to be very rare to see this in practice.

2.4 Conformance

2.4.1 Node Support

The minimum required voxel dimensions that shall be supported are 256x256x256.

2.4.2 Hardware requirements

There is no specific requirements for hardware accelaration of this component. In addition, this component does not define the specific implementation strategy to be used by a given rendering style. It is equally valid to implement the code using simple multi-pass rendering as it is to use hardware shaders.

2.4.3 Scene Graph Interaction

Sensor nodes that require interaction with the geometry (eg TouchSensor) shall provide intersection information based on the volume's bounds for minimum conformance. An implementation may optionally provide real intersection information based on performing ray casting into the volume space and reporting the first non-transparent voxel hit.

Navigation and collision detection shall also require a minimal conformance requirement of using the bounds of the volume. In addtion, the implementation may allow greater precision with non-opaque voxels, in a similar manner to the sensor interactions.

3 Abstract Types

3.1 X3DVolumeNode

```
X3DVolumeNode : X3DChildNode, X3DBoundedObject {
  SFVec3f [in,out] dimensions      1 1 1      [0,∞)
  SFNode  [in,out] metadata        NULL       [X3DMetadataObject]
  SFVec3f []       bboxCenter      0 0 0      (-∞,∞)
  SFVec3f []       bboxSize        -1 -1 -1   [0,∞) or -1 -1 -1
}
```

This abstract node type is the base type for all node types that describe volumetric data to be rendered. It sits at the same level as the polygonal *X3DShapeNode* (see ISO/IEC 19774-1 12.3.4 X3DShapeNode) within the scene graph structure, but defines volumetric data rather polygons.

The *dimensions* field specifies the dimensions of this geometry in the local coordinate space using standard X3D units. It is assumed the volume is centered around the local origin. If the bounding box size is set, it will typically be the same size as the dimensions.

3.2 X3DVolumeRenderStyleNode

```
X3DVolumeRenderStyleNode : X3DNode {
```

```
  SFBool [in,out] enabled TRUE
}
```

This abstract node type is the base type for all node types which specify a specific visual rendering style to be used.

The *enabled* field defines whether this rendering style should be currently applied to the volume data. If the field is set to FALSE, then the rendering shall not be applied at all. The render shall act as though no volume data is rendered when set to FALSE. Effectively, this allows the end user to turn on and off volume rendering of specific parts of the volume without needing to add or remove style definitions from the volume data node.

4 Node reference

4.1 BoundaryEnhancementVolumeStyle

```
BoundaryEnhancementVolumeStyle : X3DVolumeRenderStyleNode {
  SFBool      [in,out] enabled           TRUE
  SFFloat     [in,out] contribution      0.2      [0,1]
  SFColorRGBA [in,out] gradientColor     0 0 0 1  [0,1]
  SFNode      [in,out] metadata          NULL     [X3DMetadataObject]
  SFNode      [in,out] surfaceNormals    NULL     [X3DTexture3DNode]
  SFNode      [in,out] transferFunction  NULL     [X3DTextureNode]
  SFFloat     [in,out] retainedOpacity   1        [0,1]
  SFFloat     [in,out] boundaryOpacity   0        [0,∞)
  SFFloat     [in,out] opacityFactor     1        [0,∞)
}
```

Provides boundary enhancement for the volume rendering style. In this style the colour rendered is based on the gradient magnitude. Faster changing gradients (surface normals) are darker than slower changing. Areas of different density are made more visible relative to parts that are relatively constant density.

The *gradientColor* field defines the maximum colour that should be applied at the point of maximum gradient change. Where there is little or no gradient change, the colour is interpolated to transparent. The transfer function, if defined, is evaluated before applying the *gradientColor* calculation.

The *surfaceNormals* field is used to provide pre-calculated surface normal information for each voxel. If provided, this shall be used for all lighting calculations. If not provided, the implementation shall automatically generate surface normals using an implementation-specific method. If a value is provided, it shall be exactly the same voxel dimensions as the base volume data that it represents. If the dimension are not identical

then the browser shall generate a warning and automatically generate its own internal normals as though no value was provided for this field.

The output colour for this style is obtained by combining a fraction of the volume's original opacity with an enhancment based on the local boundary strength (magnitude of the gradient between adjacent voxels). Any colour information (RGB components) obtained from the transfer function evaluation is untouched by this style. The function used is

$O_g = O_v \, (k_{gc} + k_{gs}(|\Delta f|)\hat{} k_{ge})$

where

- k_{gc} is the amount of initial opacity to mix into the output (*retainedOpacity*)
- k_{gs} is the amount of the gradient enhancement to use (*boundaryOpacity*)
- k_{ge} is a power function to control the slope of the opacity curve to highlight the dataset. (*opacityFactor*)

4.2 CartoonVolumeStyle

```
CartoonVolumeStyle : X3DVolumeRenderStyleNode {
  SFBool       [in,out] enabled          TRUE
  SFNode       [in,out] metadata         NULL     [X3DMetadataObject]
  SFNode       [in,out] surfaceNormals   NULL     [X3DTexture3DNode]
  SFColorRGBA  [in,out] parallelColor    0 0 0 1        [0,1]
  SFColorRGBA  [in,out] orthogonalColor  1 1 1 1        [0,1]
  SFInt32      [in,out] colorSteps       4        [1,64]
}
```

Uses the cartoon-style nonphotorealistic rendering of the volume. Cartoon rendering uses two colours that are rendered in a series of distinct flat-shaded sections based on the local surface normal's closeness to the average normal, with no gradients in between.

The *surfaceNormals* field contains a 3D texture with at least 3 component values. Each voxel in the texture represents the surface normal direction for the corresponding voxel in the base data source. This texture should be identical in dimensions to the source data. If not, the implementation may interpolate or average between adjacent voxels to determine the average normal at the voxel required. If this field is empty, the implementation shall automatically determine the surface normal using algorithmic means.

The *parallelColor* field specifies the colour to be used for surface normals that are orthogonal to the viewer's current location (the plane of the surface itself is parallel to the user's view direction).

The *orthogonalColor* field specifies the colour to be used for surface normals that are parallel to the viewer's current location (the plane of the surface itself is orthogonal to the user's view direction). Surfaces that are further than orthgonal to the viewpoint position (ie back facing) are not rendered and shall have no colour calculated for them.

The *colorSteps* field indicates how many distinct colours should be taken from the interpolated colours and used to render the object. If the value is 1, then no colour interpolation takes place, and only the orthogonal colour is used to render the surface with. Any other value and the colours are interploted between *parallelColor* and *orthogonalColor* in HSV colour space for the RGB components, and linearly for the alpha component. From this, determine the number of colours using a midpoint calculation.

To determine the colours to be used, the angles for the surface normal relative to the view position are used. Divide the range $\pi/2$ by *colourSteps*.(The two ends of the spectrum are not interpolated in this way and shall us the specified field values). For each of the ranges , other than the two ends, find the midpoint angle and determine the interpolated colour at that point.

For example, using the default field values, the colour ranges would be used:

- 1,1,1,1 for angles $[0 - \pi/8)$
- 0.625,0.625,0.625,1 for angles $[\pi/8 - \pi/4)$,
- 0.375,0.375,0.375,1 for angles $[\pi/4 - 3\pi/8)$,
- 0,0,0,1 for angles $[3\pi/8 - \pi/2]$

4.3 CompositeVolumeStyle

```
CompositeVolumeStyle : X3DVolumeRenderStyleNode {
  SFBool [in,out] enabled  TRUE
  SFNode [in,out] metadata NULL  [X3DMetadataObject]
  SFBool [in,out] ordered  FALSE
  MFNode [in,out] styles   []   [X3DVolumeRenderStyleNode]
}
```

A rendering style node that allows compositing multiple styles together into a single rendering pass. This is used, for example to render a simple image with both edge and silhouette styles.

The *styles* field contains a list of contributing style node references that can be applied to the object. Whether the styles should be strictly rendered in order or not is dependent on the *ordered* field value. If this field value is FALSE, then the implementation may apply the various styles in any order (or even in parallel if the underlying implementation supports it). If the value is TRUE, then the implementation shall apply each style strictly in the order declared, starting at index 0.

4.4 EdgeEnhancementVolumeStyle

```
EdgeEnhancementVolumeStyle : X3DVolumeRenderStyleNode {
  SFFloat      [in,out] contribution      0.2      [0,1]
  SFColorRGBA  [in,out] edgeColor         0 0 0 1  [0,1]
  SFBool       [in,out] enabled           TRUE
  SFFloat      [in,out] gradientThreshold 0.4      [0,0-π/2]
  SFNode       [in,out] metadata          NULL     [X3DMetadataObject]
  SFNode       [in,out] surfaceNormals    NULL     [X3DTexture3DNode]
}
```

Provides edge enhancement for the volume rendering style. Enhancement of the basic volume is provided by darkening voxels based on their orientation relative to the view direction. Perpendicular voxels are coloured according to the *edgeColor* while voxels parallel are not changed at all. A threshold can be set where the proportion of how close to parallel the direction needs to be before no colour changes are made.

The *gradientThreshold* field defines the minimum angle (in radians) away from the view direction vector that the surface normal needs to be before any enhancement is applied.

The *edgeColor* field defines the colour to be used to highlight the edges.

The *surfaceNormals* field contains a 3D texture with at least 3 component values. Each voxel in the texture represents the surface normal direction for the corresponding voxel in the base data source. This texture should be identical in dimensions to the source data. If not, the implementation may interpolate or average between adjacent voxels to determine the average normal at the voxel required. If this field is empty, the implementation shall automatically determine the surface normal using algorithmic means.

4.5 ISOSurfaceVolumeStyle

```
ISOSurfaceVolumeStyle : X3DVolumeRenderStyleNode {
  SFBool [in,out] enabled        TRUE
  SFBool [in,out] lighting       FALSE
```

```
  SFNode [in,out] metadata       NULL [X3DMetadataObject]
  SFNode [in,out] surfaceNormals NULL [X3DTexture3DNode]
  SFBool [in,out] shadows        FALSE
}
```

Renders the volume using ISO-surface colouring.

The *lighting* field controls whether the rendering should calculate and apply shading effects to the visual output. When shading is applied, the value of the *surfaceNormals* field can be used to provide pre-generated surface normals for lighting calculations.

The *shadows* field controls whether the rendering should calculate and apply shadows to the visual output. A value of FALSE requires that no shadowing be applied. A value of TRUE requires that shadows be applied to the object. If the *lighting* field is set to FALSE, this field shall be ignore and no shadows generated.

The *surfaceNormals* field contains a 3D texture with at least 3 component values. Each voxel in the texture represents the surface normal direction for the corresponding voxel in the base data source. This texture should be identical in dimensions to the source data. If not, the implementation may interpolate or average between adjacent voxels to determine the average normal at the voxel required. If this field is empty, the implementation shall automatically determine the surface normal using algorithmic means.

4.6 OctTree

```
OctTree : X3DChildNode, X3DBoundedObject {
  MFNode  [in,out] highRes      NULL  [X3DChildNode]
  SFNode   [in,out]  lowRes         NULL   [X3DGroupingNode,  X3DShapeNode,
X3DVolumeShapeNode]
  SFNode  [in,out] metadata     NULL  [X3DMetadataObject]
  SFBool  [out]    lowResActive
  SFVec3f []       center       0 0 0 (-∞,∞)
  SFFloat []       range        20    [0,∞)
}
```

Allows for the definition of multiresolution data sets that resolve using octants of volume. This node is not restricted to only having volume data as its children - all other geometry types are also valid structures.

The level of detail is switched depending upon whether the user is closer or further than *range* metres from the coordinate *center*.

The *lowRes* field holds the low resolution object instance to be rendered when the viewer is outside *range* metres. The *highRes* field is used to

hold the geometry to be viewed when the inside *range* metres. An OctTree renders up to 8 children sub graphs as defined by the *highRes* field. If this field contains more than 8 children, only the first 8 shall be rendered. If less than 8 children are defined, all shall be rendered. It is up to the user to spatially located the geometry for each of the children subgraphs.

4.7 OpacityMapVolumeStyle

```
OpacityMapVolumeStyle : X3DVolumeRenderStyleNode {
  SFBool [in,out] enabled          TRUE
  SFNode [in,out] metadata         NULL [X3DMetadataObject]
  SFNode [in,out] transferFunction NULL [X3DTextureNode]
}
```

Renders the volume using the opacity mapped to a transfer function texture. This is the default rendering style if none is defined for the volume data.

The *transferFunction* field holds a single texture representation in either two or three dimensions that map the voxel data values to a specific colour output. If no value is supplied for this field, the default implementation shall generate a 256x1 greyscale alpha-only image that blends from completely opaque at pixel 0 to fully transparent at pixel 255.

4.8 SegmentedVolumeData

```
SegmentedVolumeData : X3DVolumeNode {
  SFVec3f [in,out] dimensions       1 1 1      (0,∞)
  SFNode  [in,out] metadata         NULL       [X3DMetadataObject]
  MFNode  [in,out] renderStyle      []         [X3DVolumeRenderStyleNode]
  MFBool  [in,out] segmentEnabled   []
  SFNode  [in,out] segmentedVoxels  []         [X3DTexture3DNode]
  SFVec3f []       bboxCenter       0 0 0      (-∞,∞)
  SFVec3f []       bboxSize         -1 -1 -1   [0,∞) or -1 -1 -1
}
```

Defines a segmented volume data set that allows for representation of different rendering styles for each segment identifier.

The *renderStyle* field optionally describes a particular rendering style to be used. If this field has a non-zero number of values, then the defined rendering style is to be applied to the object. If the object is segmented, then the index of the segment shall look up the rendering style at the given index in this array of values and apply that style to data described by that segment ID. If the field value is not specified by the user, the

implementation shall use a OpacityMapVolumeStyle node with default values.

The *segmentedVoxels* field holds a 3D texture with the segment IDs for each voxel. The texture supplied must have either 2 or 4 colour components. The alpha component of the texture defines the segment identifier at each voxel. If a 3-component texture is supplied, the red and green components shall define 2D data at that voxel, and the blue component shall contain the segment identifier. Providing a source texture with 1 component shall result in no rendering of this data.

The *segmentEnabled* field allows for controlling whether a segment should be rendered or not. The indices of this array corresponds to the segment ID. A value at index *i* of FALSE marks any data with the corresponding segment ID to be not rendered. If a segment ID is used that is greater than the length of the array, the value is assumed to be TRUE.

4.9 SilhouetteEnhancementVolumeStyle

```
SilhouetteEnhancementVolumeStyle : X3DVolumeRenderStyleNode {
  SFBool      [in,out] enabled           TRUE
  SFNode      [in,out] metadata          NULL     [X3DMetadataObject]
  SFNode      [in,out] surfaceNormals    NULL     [X3DTexture3DNode]
  SFFloat     [in,out] silhouetteFactor  0.5      [0,∞)
}
```

Provides silhouette enhancement for the volume rendering style. Enhancement of the basic volume is provided by darkening voxels based on their orientation relative to the view direction. Perpendicular voxels are coloured according to the *edgeColor* while voxels parallel are not changed at all. A threshold can be set where the proportion of how close to parallel the direction needs to be before no colour changes are made.

$O_g = O_v * (1 - |\mathbf{n} . \mathbf{V}|) \text{ ^ } silhouetteFactor$

The *surfaceNormals* field contains a 3D texture with at least 3 component values. Each voxel in the texture represents the surface normal direction for the corresponding voxel in the base data source. This texture should be identical in dimensions to the source data. If not, the implementation may interpolate or average between adjacent voxels to determine the average normal at the voxel required. If this field is empty, the implementation shall automatically determine the surface normal using algorithmic means.

4.10 StippleVolumeStyle

```
StippleVolumeStyle : X3DVolumeRenderStyleNode {
  SFFloat [in,out] distanceFactor              1     [0,∞)
  SFBool  [in,out] enabled                     TRUE
  SFFloat [in,out] interiorFactor              1     [0,∞)
  SFFloat [in,out] lightingFactor              1     [0,∞)
  SFNode  [in,out] metadata                    NULL  [X3DMetadataObject]
  SFFloat [in,out] gradientThreshold           0.4   [0,0-π/2]
  SFFloat [in,out] gradientRetainedOpacity     1     [0,1]
  SFFloat [in,out] gradientBoundaryOpacity     0     [0,∞)
  SFFloat [in,out] gradientOpacityFactor       1     [0,∞)
  SFFloat [in,out] silhouetteRetainedOpacity   1     [0,1]
  SFFloat [in,out] silhouetteBoundaryOpacity   0     [0,∞)
  SFFloat [in,out] silhouetteOpacityFactor     1     [0,∞)
  SFFloat [in,out] resolutionFactor            1     [0,∞)
}
```

Renders the volume using stipple patterns making use of the Wang stipple patterns for 3D dimensional data sets. Stipple patterns are automatically generated by the browser internals based on a number of algorithmic hints. It is recommended the approach defined in [STIPPLE] is used.

The general approach of the rendering process is to render a set of points, whose density is defined by a number of factors - edge, boundary silhouette enhancements, lighting and other effects. The renderer determines a absolute maximum density of points in a voxel (N_{max} and then evaluates every voxel to obtain the number of points (N) to be rendered. The distribution of points in the volume of space is an implementation- specific detail. The final calculation of N is determined by the follow set of equations:

The *gradientThreshold* field defines the minimum angle (in radians) away from the view direction vector that the surface normal needs to be before any boundary enhancement is applied.

$N = N_{max} * T_b * T_s * T_d * T_l * T_r * T_i$
$T_b = C_v * (k_{gc} + k_{gs} * (|\Delta f|)^{\wedge}k_{ge})$
$T_s = C_v * (k_{sc} + k_{ss} * (1 - (|\Delta f \,.\, \mathbf{V})) \wedge k_{se})$
$T_d = 1 + (z / a) \wedge k_{de}$
$T_l = 1 - (\mathbf{L}_i \,.\, \Delta f) \wedge k_{le}$
$T_r = ((D_{near} + d_i) / (D_{near} + d_0)) \wedge k_{re}$
$T_i = ||\Delta f|| \wedge k_{ie}$

where

- k_{gc} is the amount of initial opacity to mix into the output (*gradientRetainedOpacity*)

- k_{gs} is the amount of the gradient enhancement to use (*gradientBoundaryOpacity*)
- k_{ge} is a power function to control the slope of the opacity curve to highlight the dataset. (*gradientOpacityFactor*)
- k_{sc} is the amount of initial opacity to mix into the output (*silhouetteRetainedOpacity*)
- k_{ss} is the amount of the gradient enhancement to use (*silhouetteBoundaryOpacity*)
- k_{se} is a power function to control the slope of the opacity curve to highlight the dataset. (*silhouetteOpacityFactor*)
- k_{de} is the distance attenuation factor (*distanceFactor*)
- k_{le} is the lighting contributions factor (*lightingFactor*)
- k_{re} is the resolution enhancement contribution factor (*resolutionFactor*)
- D_{near} is the distance from the eye to the near clipping plane.
- d_i is the current distance from the near clipping plane to the volume's center.
- d_0 is the original distance from the near clipping plane to the volume's center at world startup time.
- k_{ie} is the interior enhancement contribution factor (*interiorFactor*).

4.11 ToneMappedVolumeStyle

```
ToneMappedVolumeStyle : X3DVolumeRenderStyleNode {
  SFColorRGBA [in,out] coolColor      0 0 1 0  [0,1]
  SFBool      [in,out] enabled        TRUE
  SFNode      [in,out] metadata       NULL     [X3DMetadataObject]
  SFNode      [in,out] surfaceNormals NULL     [X3DTexture3DNode]
  SFColorRGBA [in,out] warmColor      1 1 0 0  [0,1]
}
```

Renders the volume using the Gooch shading model of two-toned warm/cool colouring. Two colours are defined, a warm colour and a cool colour and the renderer shades between them based on the orientation of the voxel relative to the user. This is not the same as the basic ISO surface shading and lighting. The following colour formula is used:

$cc = (1 + \mathbf{L}_i \cdot \mathbf{n}) * 0.5$
$C_g = \Sigma\, cc * warmColor + (1 - cc) * coolColor$

The *warmColor* and *coolColor* fields define the two colours to be used at the limits of the spectrum. The *warmColor* field is used for surfaces facing towards the light, while the *coolColor* is used for surfaces facing away from the light direction.

The *surfaceNormals* field contains a 3D texture with at least 3 component values. Each voxel in the texture represents the surface normal direction for the corresponding voxel in the base data source. This texture should be identical in dimensions to the source data. If not, the implementation may interpolate or average between adjacent voxels to determine the average normal at the voxel required. If this field is empty, the implementation shall automatically determine the surface normal using algorithmic means.

4.12 VolumeData

```
VolumeData : X3DVolumeNode {
  SFVec3f [in,out] dimensions      1 1 1      [0,∞)
  SFNode  [in,out] metadata        NULL       [X3DMetadataObject]
  SFNode  [in,out] renderStyle     NULL       [X3DVolumeRenderStyleNode]
  MFNode  [in,out] voxels          []         [X3DTexture3DNode]
  SFVec3f []       bboxCenter      0 0 0      (-∞,∞)
  SFVec3f []       bboxSize        -1 -1 -1   [0,∞) or -1 -1 -1
}
```

Defines the volume information to be used on a simple non-segmented volumetric description that uses a single rendering style node for the complete volume.

The *renderStyle* field allows the user to specify a specific rendering technique to be used on this volumetric object. If the value not specified by the user, the implementation shall use a OpacityMapVolumeStyle node with default values.

The *voxels* field provides the raw voxel information to be used by the specific rendering styles. The value is any *X3DTexture3DNode* type and may have any number of colour components defined. The specific interpretation for the values at each voxel shall be defined by the value of the *renderStyle* field. If more than one node is defined for this field then each node after the first shall be treated as a mipmap level of monotonically decreasing size. Each level should be half the dimensions of the previous level

5 Support Levels

The Volume rendering component defines three levels of support as specified in Table 2.

Table 2 — Volume rendering component support levels

Level	Prequisites	Nodes/Features	Support
Level 1	Core 1 Grouping 1 Shape 1 Rendering 1		
		X3DVolumeRenderStyleNode	n/a
		X3DVolumeShapeNode	n/a
		OctTree	All fields fully supported
		OpacityMapVolumeStyle	Only 2D texture transfer function needs to be supported. All other fields fully supported.
		VolumeData	All fields fully supported
Level 2	Core 1 Grouping 1 Shape 1 Rendering 1		

		BoundaryEnhancementVolumeStyle	All fields fully supported
		CompositeVolumeStyle	ordered field is always treated as FALSE. All other fields fully supported
		EdgeEnhancementVolumeStyle	All fields fully supported
		SegmentedVolumeData	All fields fully supported
		SilhouetteEnhancementVolumeStyle	All fields fully supported
		ToneMappedVolumeStype	All fields fully supported

Level 3	Core 1 Grouping 1 Shape 1 Rendering 1		
		ISOVolumeStyle	All fields fully supported
		CartoonVolumeStyle	All fields fully supported
		CompositeVolumeStyle	All fields fully supported
		StippleVolumeStype	All fields fully supported

Appendix B

Form 3 - electronic

PROPOSAL FOR A NEW WORK ITEM	
Date of presentation of proposal:	Proposer: ISO/IEC JTC 1/SC 24 N
Secretariat: National Body BSI	ISO/IEC JTC 1 N ISO/IEC JTC 1/SC 24 N

A **proposal for a new work item** shall be submitted to the secretariat of the ISO/IEC joint technical committee concerned with a copy to the ISO Central Secretariat.

Presentation of the proposal - to be completed by the proposer

Guidelines for proposing and justifying a new work item are given in ISO Guide 26.

Title (subject to be covered and type of standard, e.g. terminology, method of test, performance requirements, etc.)

Extensible 3D (X3D) Encodings Revision

Scope (and field of application)

See Attachment

Purpose and justification - attach a separate page as annex, if necessary

See Attachment

Programme of work

If the proposed new work item is approved , which of the following document(s) is (are) expected to be developed?

	a single International Standard
	more than one International Standard (expected number: ....)
X	a multi-part International Standard consisting of ....3.... parts
	an amendment or amendments to the following International Standard(s)
	a new part to the following multi-part International Standard:
	a technical report , type ....

Relevant documents to be considered

ISO/IEC 14772, ISO/IEC 19775:2004, ISO/IEC 19776:2005, ISO/IEC 19777:2005, NP for ISO/IEC 19775:2004 revision

Cooperation and liaison

Web3D Consortium, JTC 1/SC29, SEDRIS Organization, World Wide Web Consortium

Preparatory work offered with target date(s)	Signature
See attachment	proposing SC Secretary

yes no

Will the services of a maintenance agency or registration authority be required?	X	
If yes, have you identified a potential candidate?	X	
If yes, indicate name **ISO Registration Authority for ISO Registry of Graphical Items**		
Are there any known requirements for coding?		X
If yes, please specify on a separate page		
Does the proposed standard concern known patented items?		X
If yes, please provide full information in an annex		

Comments and recommendations of the JTC secretariat - attach a separate page as annex, if necessary

Comments with respect to the proposal in general, and recommendations thereon:
It is proposed to assign this new item to JTC 1/SC 24

Voting on the proposal

Each P-member of the ISO/IEC joint technical subcommittee has an obligation to vote within the time limits laid down (normally three months after the date of circulation).

Date of circulation:	Closing date for voting:	Signature of SC secretary

Attachment to JTC 1/SC24 N 2334
Proposal for a New Work Item
Extensible 3D (X3D) Encodings Revision

Title

Extensible 3D (X3D) Encodings Revision

Scope

See clause 1.

Purpose and justification

See clause 2.

Programme of work

See clause 3.

This project shall be a multi-part International Standard, initially consisting of three parts. This multi-part International Standard is intended to be a revised version of ISO/IEC 19776:2005 as modified by Amendment 1 with the addition of new functionality.

Relevant documents to be considered

ISO/IEC 14772, ISO/IEC 19775, ISO/IEC 19776, ISO/IEC 19777

Cooperation and liaison

Web3D Consortium

JTC 1/SC 29

Preparatory Work offered with target date(s)

See clause 3.

1 Scope and Field of Application

The standards to be produced by this project specifies encodings for functionality in the revision of ISO/IEC 19775-1.

2 Purpose and Justification

2.1 Introduction

The X3D specification defined in ISO/IEC 19775-1 specifies representations whereby three-dimensional objects and behaviours can be described. X3D has been approved as a ubiquitous standard for supporting 3D content. X3D is now being revised to incorporate new functionality.

To be effectively utilized, encodings of the functionality need to be specified. This New Work Item Proposal revise ISO/IEC 19776—X3D encodings (all parts) to support the functionality being specified in revision to ISO/IEC 19775-1.

2.2 Purpose

X3D was developed to meet a specific set of market and technical requirements developed by the Web3D Consortium. To meet these requirements, X3D adopted the following design objectives:

- Separate the runtime architecture from the data encoding
- Support a variety of encoding formats, including the Extensible Markup Language (XML)
- Add new graphical, behavioural and interactive objects
- Provide alternative application programmer interfaces (APIs) within the 3D scene
- Modularize the architecture into components
- Define subsets of the specification ("Profiles") that meet different market needs
- Allow for the specification to be implemented at varying levels of service
- Eliminate, where possible, unspecified or underspecified behaviours

X3D already has a rich set of features to support applications such as engineering and scientific visualization, multimedia presentations, entertainment and educational titles, web pages, and shared virtual worlds:

- **3D graphics** - Polygonal geometry, parametric geometry, hierarchical transformations, lighting, materials, and multi-pass/multi-stage texture mapping;
- **2D graphics** - Spatialized text; 2D vector graphics; 2D/3D compositing;
- **Animation** - Timers and interpolators to drive continuous animations; humanoid animation and morphing;

- **Spatialized audio and video** - Audiovisual sources mapped onto geometry in the scene;

- **User interaction** - Mouse-based picking and dragging; keyboard input;

- **Navigation** - Cameras; user movement within the 3D scene; collision, proximity and visibility detection;

- **User-defined objects** - Ability to extend built-in browser functionality by creating user-defined data types;

- **Scripting** - Ability to dynamically change the scene via programming and scripting languages;

- **Networking** - Ability to compose a single X3D scene out of assets located on a network; hyperlinking of objects to other scenes or assets located on the World Wide Web;

- **Physical simulation** - Humanoid animation; geospatial datasets; integration with Distributed Interactive Simulation (DIS) IEEE 1278.1 protocols,

- **Programmable shaders** – Support for programmable shading languages to that authors can take maximum advantage of modern 3D hardware as well as create the effects needed for their purposes;

- **3D and Cube Map Textures** – Includes the ability to use volumetric and environment textures;

- **Improved LOD node** – Adds an output event whenever a level change takes place;

- **Improved text support** – Provides information that allows access to text string extent information; and

- **Improved CAD support** – Adds a base CAD profile and features designed to support basic interchange of CAD data.

New functionality is being proposed to complement that described above. This new functionality will include such items as the following:

- **Layers** – Allows grouping scenes into several separate subscenes each with its own transformation hierarchy. The order of processing of the subscenes can be specified.
- **Compositing** – Allows specifying rendering of scenes to alternate locations and use of scenes created by other applications within X3D worlds.
- **Volume rendering** – Provides support for the direct rendering of volume images such as those created by CAT scans.
- **Picking sensors** – Adds sensors that return the identification of objects selected.

- **DIS extensions** – Generalizes the DIS component to widen the scope of applications, making the component generally useful for communication across the Internet.
- **Updated GeoSpatial functionality** – Improves the transformability of objects that require GeoSpatial positioning as well as adding GeoSpatial-knowledgeable touch sensors.
- **Ortho viewpoints** – Adds support for orthographic projections to augment the currently available perspective projections.
- **Physics nodes** – Allows including knowledge of rigid-body physics characteristics to objects in an X3D scene
- **Non-linear interpolators** – Augments the current linear interpolators with the ability to provide general non-linear interpolation.

This NP supports projects to define encodings for all of the above functionality with the intention of providing as much backwards compatibility as possible with the existing specification of ISO/IEC 19776.

2.3 Relationship to existing standards

Organization	Standard	Relationship
JTC 1 SC24/WG6	X3D	ISO/IEC 19775 specifies a complete and powerful technology for representing and interacting with dynamic 3D data including support for profiles that make tailoring conformant 3D worlds to special needs possible. Part 1 specifies the base functionality for X3D. Part 2 specifies the Scene Access Interface (SAI) that specifies a consistent API for internal scripting and external access. Amendment 1 to Part 1 added new functionality and clarifications.
JTC 1 SC24/WG6	X3D encodings	ISO/IEC 19776 specifies the encodings for X3D. These include XML (Part 1), Classic VRML (Part 2), and Binary (Part 3—FCD text in preparation). Amendments to Parts 1 and 2 have been processed to support the new nodes introduced by Amendment 1 to ISO/IEC 19775. Part 3 is being processed to include the new nodes introduced by Amendment 1 to ISO/IEC 19775.
JTC 1 SC24/WG6	X3D language bindings	ISO/IEC 19777 specifies bindings for the X3D SAI to programming languages. Part 1 specifies a binding to ECMAScript. Part 2 specifies a binding to Java.
JTC 1 SC24/WG6	VRML	ISO/IEC 14772 is the specification for VRML. X3D is supplanting the VRML standard with functionality inclusive of ISO/IEC 14772 and profiles to provide compatibility with VRML content.
JTC 1 SC29/WG11	MPEG-4	MPEG-4 has normatively referenced VRML as the basis for BIFS. MPEG-4 has also accepted X3D's Interactive Profile for the basis of MPEG-4 3D Interactive Profile
World Wide Web Consortium (W3C)	XML	XML is the basis for the XML encoding of X3D, as well as a family of related technologies including the Document Object Model (DOM), XML Schema and Extensible Stylesheet Language for Transformations (XSLT). XML is based on SGML but differs in some aspects. XML has become the basis for data interchange on the WWW.

2.4 Justification

The industry has made a significant investment in developing 3D content with VRML. However, there have also been major advances in graphics hardware and Internet technologies since VRML was standardized in 1997, and a continuing specter of fragmentation into proprietary implementations because VRML has not kept pace with these advances. X3D addressed this feature gap as well as some longstanding interoperability and conformance issues with VRML, and provides an interoperability framework for all future 3D content. Now additional advances in technology have demonstrated the need for additional functionality to be included in X3D. Due to the complexity and diversity of this functionality, it is thought that a revision to the original X3D standard incorporating the changes specified in amendment 1 to that standard is the best approach for smoothly integrating all aspects of X3D functionality in a manner that ensures smooth integration in the X3D framework while maximizing backwards compatibility for existing X3D content and implementations. Coincident with this, encodings are needed for access to X3D capabilities.

2.5 Feasibility

Many of the new features in X3D have existed as either X3D or VRML-based prototypes or extensions to popular browsers and have been proven in a production setting. These prototypes will be validated and tested for feasibility and interoperability prior to CD. It is intended that this revision of X3D demonstrate a high-degree of compatibility with existing X3D functionality while smoothly adding the necessary new capabilities. These new capabilities will have been implemented at least twice prior to their inclusion in FCD text.

2.6 Timeliness

X3D was approved in 2004 after a continuous development cycle that was initiated in 2000. X3D improved upon VRML by incorporating many of the advancements in graphics hardware and Internet technology that have emerged since that time. X3D developers using existing commercial authoring tools are already capable of creating rich 3D content that exploits these new features. X3D has provided the production pipeline for deploying that content over the Internet. Now, as these commercial authoring tools provide greater capabilities, X3D improvements are needed to efficiently migrate these new capabilities to the World Wide Web.

2.7 Urgency

If this standard is not developed, similar capabilities will be added in non-interoperable ways through the use of proprietary 3D run-time systems. The rapid growth of interoperable web- and broadcast-based 3D will be compromised without delivering a new X3D standard that supports these new capabilities.

2.8 Benefits

It is expected that this standard will increase the productivity of those involved in the development of Internet-based applications. The sound technical base of X3D coupled with the established success of VRML provides a low-risk solution to meeting a large need among 3D content providers. Other benefits include improved performance and graphics quality for Internet users, resulting in more compelling Internet information enhanced by the three-dimensional capabilities of X3D.

2.9 Risk identification and mitigation

The key risks to the timely and successful development of X3D are:

a) extended development—the standard must be developed within a specified time frame or the feasibility of continuing the work must be reexamined;

b) overlap with external activities, especially external standards activities with their own (already fixed) scopes and schedules. Coordinated interoperability with non-SC24 standards, and Internet-related standards in particular, will be a long term goal of X3D;

c) unproven technologies—as stated earlier, the performance of 3D scene graphs may be too demanding for DOM integration as a common animation mechanism, though simple web-page integration and offline authoring will be valuable nonetheless.

To mitigate these risks, the following risk reduction strategies will be used:

d) joint development with the original designer of the X3D standard (Web3D Consortium). This project will operate under the provisions of a Cooperative Working Agreement between SC24 and the Web3D Consortium as already approved by the ISO/IEC Council;

e) close coordination with appropriate standards committees and user groups;

f) maximum feasible upwards compatibility of any functional changes to that in the existing prototype implementation;

g) separation of the speculative technologies into their own part of the standard, to reduce the potential impact on the overall specification;

h) An extensive list of joint technical efforts related to XML-ization of the X3D specification is under consideration by the Web3D and W3C consortia.

3 Programme of work / schedule

The following schedule is proposed by SC24 and the Web3D Consortium who will work cooperatively to create and review both this NP and the CD text that will accompany it:

• Draft NP	SC24	2/2006
• Initiate NP ballot	SC24	2/2006
• Produce CD text and Initiate CD ballot	SC24/Web3D Consortium	2/2006
• Editing Meeting to prepare FCD text	SC24/Web3D Consortium	6/2006
• Initiate FCD ballot	SC24	8/2006
• Editing Meeting to prepare FDIS text	SC24/WG6 & Web3D Consortium	1/2007
• Submit FDIS text for balloting	JTC1	2/2007
• Publish IS	ITTF	6/2007

NEW WORK ITEM PROPOSAL - PROJECT ACCEPTANCE CRITERIA				
Criterion		**Validity**		**Explanation**
A	**Business Requirement**			
A.1	Market Requirement	Essential	X	The VRML standard ISO/IEC 14772 defines the semantics and UTF-8 encoding for a file format which has become the standard way of representing three-dimensional information on the World Wide Web. The Web3D Consortium (formerly "VRML Consortium") desires to establish X3D as the successor to the VRML standard. X3D improved upon VRML with new features, advanced application programmer interfaces, additional data encoding formats, stricter conformance, and a componentized architecture that allows for a modular approach to supporting the specification. X3D provides compatibility with VRML content via its componentized architecture and the use of profiles. X3D is used on a variety of hardware devices and in a broad range of application areas such as engineering and scientific visualization, multimedia presentations, entertainment and educational titles, web pages, and shared virtual worlds. X3D also serves as a universal interchange format for integrated 3D graphics and multimedia. All of these applications have already been demonstrated. By working cooperatively with the Web3D Consortium to augment the X3D International Standard with new required functionality, the long term stability and future growth of web- and broadcast-based 3D will be enhanced. The technologies upon which X3D is based are already well established within the VRML and X3D communities and the computer graphics industry at large. By revising the X3D International Standard, the interoperability of the Internet with other graphics and communications technologies (such as those recommended by the ITU-T) will be enhanced. This NP establishes projects to define the encodings for the X3D revision.
		Desirable		
		Supportive		
A.2	Regulatory Context	Essential		
		Desirable		
		Supportive		
		Not Relevant	X	X3D is subject to no known Regulatory requirements.

B.	**Related Work**			
B.1	Completion/Maintenance of current standards	Yes	X	ISO/IEC 19775:2004, and consequently ISO/IEC 19776:2005, is a widely used and successful standard. To continue its effectiveness, it is necessary to augment its capabilities to support requirements of applications that are beginning to appear in the industry.
		No		
B.2	Commitment to other organization	Yes	X	This NP represents a revision of a series of International Standard based on work initiated within the Web3D Consortium and transposed with the cooperation of JTC 1/ SC 24. JTC 1/ SC 24 and the Web3D Consortium have already agreed to cooperate in this work and a Cooperative Agreement approved by both organizations has been approved by ISO/IEC Council.
		No		
B.3	Other sources of standards	Yes		
		No	X	The directions for this question state that it addresses other known activities that "might be available to JTC 1 as PAS." Since this is the continuation of work on a series of standards developed in cooperation with the Web3D Consortium, neither JTC 1/ SC 24 nor the Web3D Consortium believes that the PAS process should be used for this work.
C.	**Technical Status**			
C.1	Mature Technology	Yes	X	The base VRML technologies underlying the X3D system have been proven to be an essential part of web-based 3D content development. No sophisticated technologies are needed for implementation. Many of the new technologies developed for X3D have been proven in production for several years. Some of the new technologies are proven but exist only in prototype form. Interoperability concerns with the various prototype implementations are being addressed and will be resolved prior to CD. All other aspects of X3D have proven to be both useful and effective. No new encoding technologies are anticipated.
		No		
C.2	Prospective Technology	Yes		
		No	X	This work is not anticipatory. The need is proven.
C.3	Models/Tools	Yes		
		No	X	This NP does not relate to the creation of supportive reference models or tools.
D.	**Other Justification**			

Notes to Proforma

A. **Business Relevance**. That which identifies market place relevance in terms of what problem is being solved and or need being addressed.

A.1. **Market Requirement**. When submitting a NP, the proposer shall identify the nature of the Market Requirement, assessing the extent to which it is essential, desirable or merely supportive of some other project.

A.2 **Technical Regulation**. If a Regulatory requirement is deemed to exist - e.g. for an area of public concern e.g. Information Security, Data protection, potentially leading to regulatory/public interest action based on the use of this voluntary international standard - the proposer shall identify this here.

B. **Related Work.** Aspects of the relationship of this NP to other areas of standardization work shall be identified in this section.

B.1 **Competition/Maintenance**. If this NP is concerned with completing or maintaining existing standards, those concerned shall be identified here.

B.2 **External Commitment**. Groups, bodies, or fora external to JTC1 to which a commitment has been made by JTC for cooperation and or collaboration on this NP shall be identified here.

B.3 **External Std/Specification**. If other activities creating standards or specifications in this topic area are known to exist or be planned, and which might be available to JTC1 as PAS, they shall be identified here.

C. **Technical Status.** The proposer shall indicate here an assessment of the extent to which the proposed standard is supported by current technology.

C.1 **Mature Technology**. Indicate here the extent to which the technology is reasonably stable and ripe for standardization.

C.2 **Prospective Technology**. If the NP is anticipatory in nature based on expected or forecasted need, this shall be indicated here.

C.3 **Models/Tools**. If the NP relates to the creation of supportive reference models or tools, this shall be indicated here.

D. **Any other aspects** of background information justifying this NP shall be indicated here.